

Design of Novel Algorithm and Architecture for Gaussian Based Color Image Enhancement System for Real Time Applications

M. C Hanumantharaju¹, M. Ravishankar¹ and D. R Rameshbabu²

¹Department of Information Science & Engineering,

²Department of Computer Science & Engineering

Dayananda Sagar College of Engineering, Bangalore, India.
(mchanumantharaju, ravishankarmcn, bobrammysore@gmail.com)

No Institute Given

Abstract. This paper presents the development of a new algorithm for Gaussian based color image enhancement system. The algorithm has been designed into architecture suitable for FPGA/ASIC implementation. The color image enhancement is achieved by first convolving an original image with a Gaussian kernel since Gaussian distribution is a point spread function which smoothes the image. Further, logarithm-domain processing and gain/offset corrections are employed in order to enhance and translate pixels into the display range of 0 to 255. The proposed algorithm not only provides better dynamic range compression and color rendition effect but also achieves color constancy in an image. The design exploits high degrees of pipelining and parallel processing to achieve real time performance. The design has been realized by RTL compliant Verilog coding and fits into a single FPGA with a gate count utilization of 321,804. The proposed method is implemented using Xilinx Virtex-II Pro XC2VP40-7FF1148 FPGA device and is capable of processing high resolution color motion pictures of sizes of up to 1600×1200 pixels at the real time video rate of 116 frames per second. This shows that the proposed design would work for not only still images but also for high resolution video sequences.

Keywords: Gaussian color image enhancement, Serpentine memory, 2D Gaussian convolution, Logarithm, Field Programmable Gate Array.

1 Introduction

Digital image enhancement [1] refers to accentuation, sharpening of image features such as edges, boundaries or contrast to make a graphic display more useful for display and analysis. The enhanced images with better contrast and details are required in many areas such as computer vision, remote sensing, dynamic scene analysis, autonomous navigation and medical image analysis. In the recent

years, color image enhancement has been becoming an increasingly important research area with the widespread use of color images. Numerous methods [2] are available in the literature for color image enhancement. The color image enhancement can be classified into two categories according to the color space: color image enhancement in RGB color space and color image enhancement based on transformed space.

The paper is organized as follows: Section 2 gives a brief review of previous work done. Section 3 describes the proposed Gaussian based color image enhancement algorithm. Section 4 provides detailed system architecture for hardware realization of color image enhancement algorithm. The results and discussions follow these. Conclusion arrived at are presented in the final section.

2 Related Work

Digital Signal Processors (DSPs) [3][4] have been employed for enhancement of images which provides some improvement compared to general purpose computers. Only marginal improvement has been achieved since parallelism and pipelining incorporated in the design are inadequate. This scheme uses optimized DSP libraries for complex operations and does not take full advantage of inherent parallelism of image enhancement algorithm. The neural network based learning algorithm [5] provides an excellent solution for the color image enhancement with color restoration. The hardware implementation of these algorithms parallelizes the computation and delivers real time throughput for color image enhancement. However, its window related operations such as convolution, summation and matrix dot products in an image enhancement architecture demands enormous amount of hardware resources.

Hiroshi Tsutsui et al. [6] proposed an FPGA implementation of adaptive real-time video image enhancement based on variational model of the Retinex theory. The authors have claimed that the architectures developed in this scheme are efficient and can handle color picture of size 1900×1200 pixels at the real time video rate of 60 frames per sec. The authors have not justified how high throughput has been achieved in spite of time consuming iterations to the tune of 30. Abdullah M. Alsuwailem et al. [7] proposed a new approach for histogram equalization using FPGAs. Although efficient architectures were developed for histogram equalization, the restored images using this scheme are generally not satisfactory.

An efficient architecture for enhancement of video stream captured in non-uniform lighting conditions was proposed by Ming Z. Zhang et al. [8]. The new architecture processes images and streaming video in the HSV domain with the homomorphic filter and converts the result back to HSV. This leads to an additional computational cost and, the error rate is high for the RGB to HSV conversion process. Digital architecture for real time video enhancement based on illumination reflection model was proposed by Hau T. Ngo et al. [9]. This scheme improves visual quality of digital images and video captured under insufficient and non-uniform lighting conditions. Bidarte et al. [10] proposed spatial-based

adaptive and reusable hardware architecture for image enhancement. However, the histogram modification used in this scheme treats all regions of the image equally and often results in poor local performance, which in turn affects the image details. The modified luminance based multiscale retinex algorithm proposed by Tao et al. [11] achieves optimal enhancement result with minimal complexity of hardware implementation. However, the algorithm works fine so long as the background is dark and the object is bright.

The limitations mentioned earlier are overcome in the proposed method in an efficient way. To start with, the input image is convolved with 5×5 Gaussian kernel in order to smooth the image. Further, the dynamic range of an image is compressed by replacing each pixel with its logarithm. In the proposed method, the image enhancement operations are arranged in an efficient way adding true color constancy at every step. It has less number of parameters to specify and provides true color fidelity. In addition, the proposed algorithm is computationally inexpensive. In the proposed scheme, an additional step is necessary to solve the gray world violation problem as is the case with the implementation reported in Ref. [11].

In the present work, in order to test the developed algorithm, standard test images have been used and results are favorably compared with that of other researchers. In order to evaluate the performance of the proposed algorithm, the metric Peak Signal to Noise Ratio (PSNR) has been used.

3 Proposed Gaussian Based Image Enhancement Algorithm

In order to account for the smoothness, lightness, color constancy and dynamic range properties of Human Visual System (HVS), Gaussian based image enhancement algorithm has been proposed. The basic operational sequence of the proposed Gaussian based color image enhancement algorithm is shown in Fig. 1. To start with, the original image (which is of poor quality and needing enhancement) is read in RGB color space. The color components are separated followed by the selection of window size as 5×5 for each of the R, G and B components. In each color component, the selected window is convolved with 5×5 Gaussian kernel in order to smooth the image. Next, Logarithmic operation is accomplished in order to compress the dynamic range of the image. Finally, Gain/Offset adjustment is done in order to translate the pixels into the display range of 0 to 255. The separated R, G and B components are combined into composite RGB in order to obtain the enhanced image.

3.1 Convolution Operation

Convolution is a simple mathematical operation which is essential in many image processing algorithms. The color image enhancement algorithm proposed in this paper uses 5×5 pixel window, where a block of twenty five pixels of original

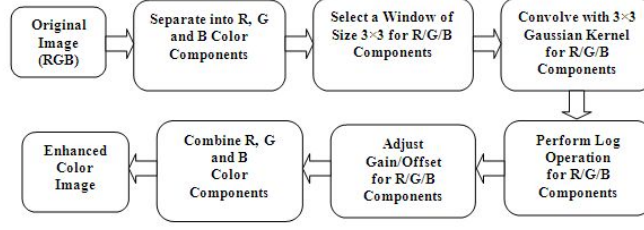


Fig. 1. Flow Sequence of the Proposed Gaussian based Color Image Enhancement Algorithm

image is convolved with Gaussian kernel of size 5×5 . The two dimension (2D) Gaussian function is defined by Eqn. (1):

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

where σ is the standard deviation of the distribution; x and y are spatial coordinates.

The basic principle of Gaussian filters is to use 2D distribution as a point spread function. This property is adopted in image enhancement application and is achieved by convolving the 2D Gaussian distribution function with the original image. A Gaussian kernel can be a 3×3 or a 5×5 matrix as shown in Fig. 2.

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Fig. 2. Gaussian Kernels: 3×3 kernel and 5×5 kernel

The Gaussian convolution matrix is given by Eqn. (2).

$$G(x, y) = I(x, y) \otimes g(x, y) \quad (2)$$

where \otimes denotes convolution, $g(x, y)$ is a Gaussian kernel, $I(x, y)$ is the original image and $G(x, y)$ is the convolved output.

Mathematically, 2D convolution can be represented by the following Eqn. (3):

$$G(x, y) = \sum_{i=1}^M \sum_{j=1}^N I(i, j) \times g(x - i, y - j) \quad (3)$$

where $m = 5$ and $n = 5$ for 5×5 Gaussian kernel. In this work, the convolution mask is chosen as 5×5 in order to enhance the implementation speed, at the same time minimizing blocking artifacts. The convolution operation for a mask of 5×5 is given by Eqn. (4)

$$P(x, y) = \frac{\sum_{i=0}^4 W_i \times P_i}{\sum_{i=0}^4 W_i} \quad (4)$$

where W_i indicates the 5×5 Gaussian mask, P_i is the 5×5 sliding window in the input image and P is the Gaussian convolved pixel. As an example, Fig. 3 illustrated the hardware implementation of convolution with mask as 3×3 . The convolution mask of 5×5 implemented in this work is similar to that shown in Fig. 3.

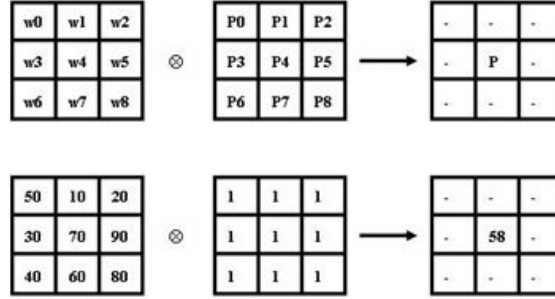


Fig. 3. Two Dimensional Convolution Scheme

Next, the log transformation operation applied on the image compresses the dynamic range of gray level input values to manageable levels for further processing. The logarithmic processing on a 2D image is carried out by using Eqn. (5):

$$G_L(x, y) = K \times \log_2 [1 + G(x, y)] \quad (5)$$

where K is a constant arrived as 1.5 after conducting a number of experiments with various test images. Further, the logarithm computed image is scaled by a scaling factor of 32 for getting pixel values in the range of 0 to 255. The final step in proposed approach is gain/offset correction. This gain/offset correction is accomplished by using Eqn. (6).

$$I'(x, y) = \frac{d_{max}}{G_{Lmax} - G_{Lmin}} [G_L(x, y) - G_{Lmin}] \quad (6)$$

where d_{max} is the maximum intensity which is chosen as 255 for an image with 8-bit representation, $G_L(x, y)$ is the log-transformed image, G_{Lmin} is the minimum value of log transformed image, G_{Lmax} is the maximum value of log

transformed image, $I'(x, y)$ is the enhanced image and, x and y represent spatial coordinates.

4 Architecture of Proposed Color Image Enhancement System

This section presents the architectures of the proposed color image enhancement system. It is composed of several components such as serpentine memory, 2D convolution, logarithm and gain/offset corrections.

4.1 System Overview

The block diagram of a top level Gaussian based color image enhancement system shown in Fig. 4, which performs a color image enhancement operation as implied in the name. The detailed signal descriptions of this block are provided in Table I. The inputs "rin", "gin", and "bin" are the red, green, and blue pixels, each of size 8-bits. The input pixels are valid at the positive edge of the clock. The outputs "ro", "go" and "bo" represent the enhanced pixels of proposed image enhancement system. The "pixel_valid" signal is asserted, When the enhanced pixel data is valid. The enhancement system can be reset at any point of time by asserting the asynchronous, active low signal, "reset_n".

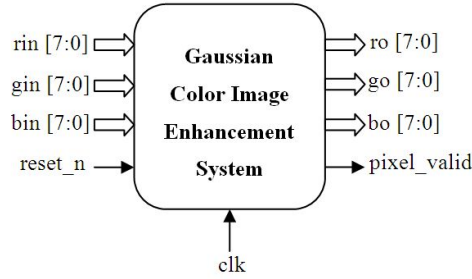


Fig. 4. Block Diagram of Top Level Gaussian Based Color Image Enhancement System

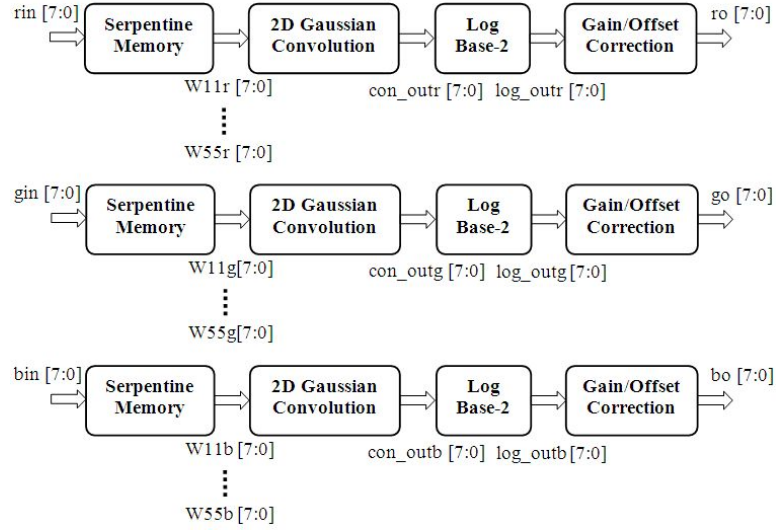
The architecture proposed for color image enhancement system consists of twelve modules, each color component (R/G/B) comprising four basic modules: Sliding window, 2D Gaussian Convolution, Logarithm Base-2, and Gain/Offset Correction as shown in Fig. 5. Pipelining and parallel processing techniques have been adopted in the design in order to increase the processing speed of the proposed system.

4.2 Serpentine Memory Architecture

The signal diagram of a serpentine memory for each of the color components R, G and B is presented in Fig. 6(a). The serpentine memory is commonly referred

Table 1. Signal Description for the Top Module of Gaussian Based Image Enhancement System

Signals	Description
clk	This is the global clock signal
reset_n	Active low system reset
rin [7:0]	Red color component
gin [7:0]	Green color component
bin [7:0]	Blue color component
ro [7:0]	Enhanced red color component
go [7:0]	Enhanced Green color component
bo [7:0]	Enhanced Blue color component
pixel_valid	Valid signal for enhanced RGB pixel

**Fig. 5.** Architecture of Gaussian Based Color Image Enhancement System

to as a sliding window. The proposed method uses 5×5 sliding window due to its ease of implementation. The pixel values of the input image "pixel_in" of width 8-bits are imported serially into the sliding window module on the positive edge of clock, "clk". Detailed hardware architecture for 5×5 sliding window function or serpentine memory that uses row buffers is shown in Fig. 6(b).

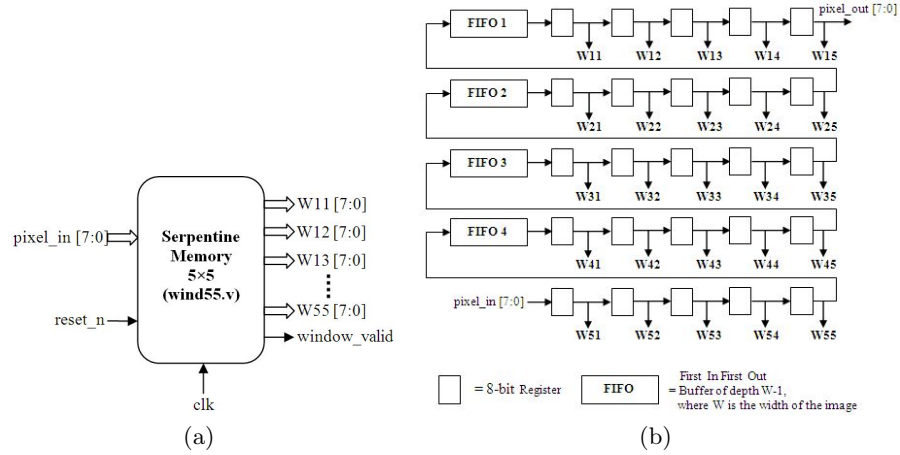


Fig. 6. (a) Signal Diagram of Serpentine Memory for R/G/B Color Channels (b) Schematic Diagram of a 5×5 Serpentine Memory for R/G/B Color Channels

In this work, one pixel is read from memory in one clock cycle with no latency. The pixels are read row by row in a raster scan order. For a 5×5 sliding window, four First-In-First-Out (FIFO) buffers are used. The FIFO buffers are used to reduce the memory access to one pixel per clock cycle. The depth of the FIFO buffer is chosen as $(W-3)$, where W is the width of the image.

4.3 Architecture of Gaussian Convolution Processor

The design of the 2D Gaussian convolution processor in hardware is more difficult than that of the sliding window. The convolution algorithm uses adders, multipliers and dividers in order to calculate its output. The signal diagram of the 2D Gaussian convolution processor is shown in Fig. 7.

The input pixels W11 to W55 are scaled to 16 bits in order to match the Gaussian coefficients. The input pixels are valid on the positive edge of the "clk" with "window_valid" asserted. The coefficient values of the convolution mask employed in most of the image processing applications remain constant for the entire processing. Therefore, constant coefficient multiplier is employed in order to implement convolution efficiently. The twenty five Gaussian coefficients G11 [15:0] to G55 [15:0] represent the convolution mask shown earlier in Fig. 2.

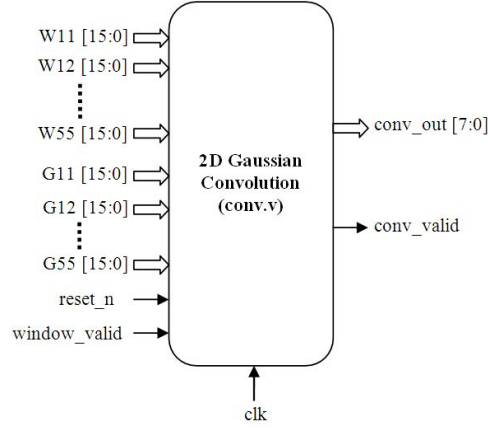


Fig. 7. Signal Diagram of 2D Gaussian Convolution Processor for R/G/B Color Channels

The output signal "conv_out" is 8-bits in width and is valid only on the positive edge of signal, "conv_valid".

4.4 Logarithm of Base-2 Architecture

The signal diagram of logarithm base-2 module for R/G/B color channels is shown in Fig. 8(a). The convolution output comprises the input for the logarithm block. The architecture for logarithmic module is shown in Fig. 8(b).

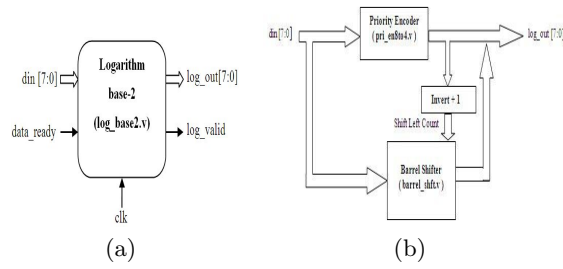


Fig. 8. (a) Signal Diagram of Logarithm Base-2 Module for R/G/B Color Channels
(b) Architecture of Logarithm Base-2 Processor for R/G/B Color Channels

The log of base-2 is computed in two steps: Integer portion computation and fractional portion computation. Accordingly, the architecture consists of two modules.

4.5 Gain/Offset Correction

The gain/offset correction is accomplished using Eqn. (6). The hardware implementation of gain and offset correction uses a multiplier and subtractor module. This step steers pixels into the correct display range: 0 to 255. The multiplier design presented in this work incorporates a high degree of parallel circuits and pipelining of five levels. The multiplier performs the multiplication of two 8-bits unsigned numbers $n1$ and $n2$ as shown in Fig. 9(a). The multiplier result is of width 16-bits. The detailed architecture for the multiplier is shown in Fig. 9(b).

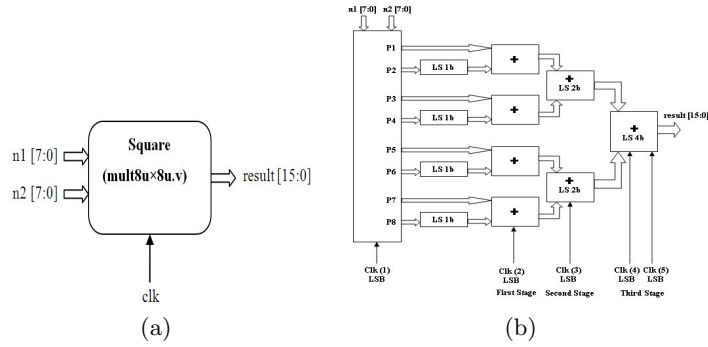


Fig. 9. (a) Signal Diagram of a Multiplier Block for R/G/B Color Channels (b) Detailed Architecture of Pipelined Multiplier Design for R/G/B Color Channels

5 Results and Discussions

Over the years, many image enhancement techniques have been reported, major ones being Multiscale Retinex with Color Restoration. The Multiscale Retinex is the most popular technique that works well under most lighting conditions. In this paper, the most popular image enhancement techniques namely, histogram equalization[12], MSRCR [13] and improved MSRCR [14] are chosen in order to validate the proposed scheme.

The image enhancement algorithms mentioned earlier have been coded in Matlab by the present authors and the reconstructed images are shown in Fig. 10. The original image of resolution 256×256 pixels is shown in Fig. 10(a). The image enhancement using conventional histogram equalization of R, G and B channels is shown in Fig. 10(b). The histogram equalization method improves the global contrast of the original image. However, the overexposed regions of the original image are highly enhanced in this approach. Image enhancement based on Multiscale Retinex with Color Restoration is shown in Fig. 10(c). The enhanced image obtained by MSRCR is better compared with histogram equalization. MSRCR method fails to produce good color rendition for a class of images that contain violation of gray world assumption.

The image enhanced based on improved multiscale retinex using hue-saturation-value (HSV) color space shown in Fig. 10(d) improves the visual quality better than MSRCR method. However, this approach is complex from computation point of view. The limitations expressed in the above algorithms are conspicuous by its absence in the proposed Gaussian based color image enhancement method as can be seen from the result displayed in Fig. 10(e). The proposed algorithm outperforms the other image enhancement techniques in terms of quality of the reconstructed picture. The images enhanced by our method are clearer, more vivid, and more brilliant than that achieved by other researchers.

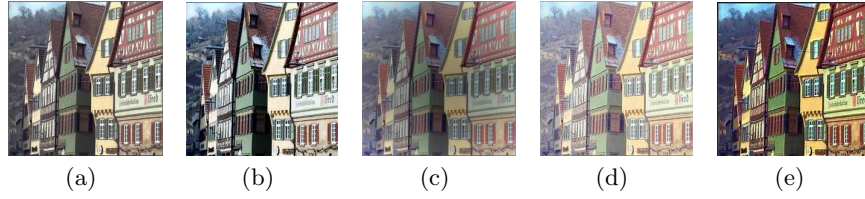


Fig. 10. Comparison of Reconstructed Pictures Using Image Enhancement Algorithms: (a) Original Image of resolution 256×256 pixels (b) Image enhanced based on Histogram Equalization of Ref. [12], PSNR: 31.2 dB (c) Multiscale Retinex with Color Restoration of Ref. [13], PSNR: 30 dB (d) Improved Multiscale Retinex in HSV Color Space of Ref. [14], PSNR: 29.5 dB (e) Proposed Gaussian Color Image Enhancement, PSNR: 31.8 dB.

The proposed FPGA implementation of Gaussian based Color Image Enhancement System has been coded and tested in Matlab (Version 8.1) first in order to ensure the correct working of the algorithm. Subsequently, the complete system has been coded in Verilog HDL so that it may be implemented on an FPGA or ASIC. The proposed scheme has been coded in RTL compliant Verilog and the hardware simulation results are compared with Matlab results described earlier in order to validate the hardware design. The system simulation has been done using ModelSim (Version SE 6.4) and Synthesized using Xilinx ISE 9.2i. The algorithm has been implemented on Xilinx Virtex-II Pro XC2VP40-7FF1148 FPGA device. In the proposed work, window size is chosen as 5×5 since the enhanced image looks more appealing than that for other sizes.

Elaborate experiments were conducted on over a dozen varieties of images and consistently good results have been obtained for the proposed Verilog implementation. As examples, three poor quality images have been enhanced using the Gaussian based color image enhancement hardware system and is presented in Fig. 11. Test images (a), (d) and (g) of Fig. 11 show the original image of resolution 256×256 pixels. The images (b), (e) and (h) of Fig. 13 show the enhanced images using the proposed method based on Matlab approach. The images in Fig. 11 (c), (f) and (i) show the enhanced images using the proposed Gaussian method based on hardware (Verilog) approach.

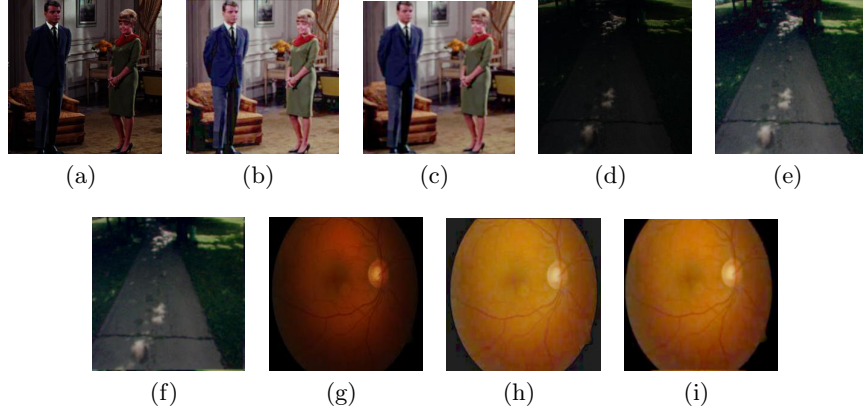


Fig. 11. Experimental Results for Gaussian based Color Image Enhancement System: (a) Original couple image (256×256 pixels) (b) Reconstructed couple image using Matlab, PSNR: 36.9 dB (c) Reconstructed couple image using Verilog, PSNR: 36.2 dB. (d) Original dark road image (256×256 pixels) (e) Reconstructed dark road image using Matlab, PSNR: 40.8 dB (f) Reconstructed dark road image using Verilog, PSNR: 40.3 dB. (g) Original eyeball image (256×256 pixels) (h) Reconstructed eyeball image using Matlab, PSNR: 36.8 dB (i) Reconstructed eyeball image using Verilog, PSNR: 36.1 dB

5.1 Verilog Simulation Results Using Modelsim

The ModelSim simulation waveforms for inputting the image pixel data for all the three color components is shown in Fig. 12(a). The enhancement process starts when the "din_valid" signal is asserted at the positive edge of the clock. The RGB image is separated into R, G and B data and the algorithm is applied concurrently to all the color components. The reconstructed pixels are issued at the output pins "dout" with "dvm" signal asserted as presented in Fig. 12(b). The outputs are issued out continuously one pixel every clock cycle after a latency of 535 clock cycles.

5.2 Place and Route Results

The design was Synthesized, Placed and Routed using Xilinx ISE 9.2i. The FPGA target device chosen was Xilinx Virtex-II Pro XC2VP40-7FF1148. The synthesis and place and route results for the design are presented in the following. The ISE generated RTL view of the top level module "Gaussian.IE" is shown in Fig. 13(a) and the zoomed top view modules are presented in Fig. 13(b). The detailed zoomed view of R/G/B processor is shown in Fig. 14.

The device utilization and the performance summary for Gaussian based color image enhancement system are presented in Fig. 15. The system utilizes about 321,804 gates as reported by the ISE tool. The maximum frequency of operation is 224.840 MHz. This works out to a frame rate of 117 per second

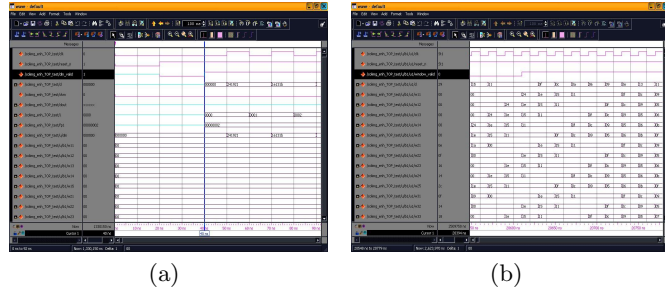


Fig. 12. (a) Waveforms for Gaussian Based Color Image Enhancement System: In-putting Image Data (b) Waveforms for Gaussian based Color Image Enhancement System: Enhanced Pixel Data

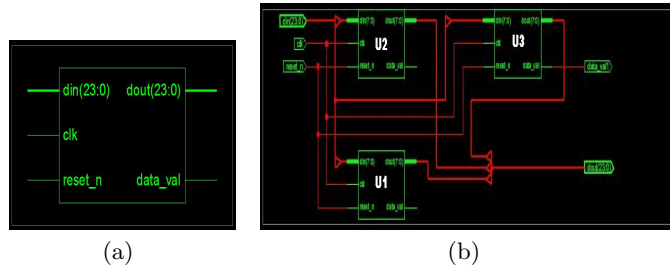


Fig. 13. Experimental Results for Gaussian based Color Image Enhancement System: (a) RTL View of the Top Module "Gaussian.IE" (b) Zoomed View of "Gaussian.IE" Module where **U1**: Red Color, **U2**: Green Color and **U3**: Blue Color Component Processors

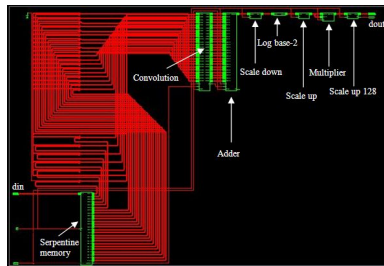


Fig. 14. Zoomed View of U1 or U2 or U3 Module.

for a picture size of 1600×1200 pixels since the design has the capability of processing one pixel every clock cycle ignoring initial latency of 535 clock cycles. As a result, it can work on any FPGA or ASIC without needing to change any code. As ASIC, it is likely to work for higher resolutions beyond 4K format at real time rates. This shows that the design would work for not only still images but also for high resolution video sequences.

Project File:		Current State:	Processing File Generated	
Gaussian_E_ice		Errors	No Errors	
colimg_vhdl_1107		Warnings	13396 Warnings	
xc2vp40-7ff1148		Updated	Tue Jan 22 09:55:22 2012	
Product Version:		ISE 9.2		
Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Notes
Number of slices 3g1:Flt00	29,337	30,784	95%	
Number of 4 input LUTs	1,432	30,784	5%	
Logic Distribution				
Number of occupied slices	15,423	15,362	78%	
Number of slices containing only related logic	15,423	15,423	100%	
Number of slices containing unrelated logic	0	15,423	0%	
Total Number of 4 input LUTs	2,047	30,784	7%	
Number used as logic	1,432			
Number used as a route thru	249			
Number used as SRAM registers	1,077			
Number of bonded I/Os	51	804	6%	
IOB flip-flops	40			
Number of FPGAs	0	2	0%	
Number of GCLs	1	96	0%	
Number of GTs	0	12	0%	
Number of GTBs	0	0	0%	
Total registered gate count for design	321,804			
Additional JTAG gate count for IOB:	2,449			

Fig. 15. FPGA Resource Utilization for Gaussian Based Color Image Enhancement Design

The timing summary for the design as reported by Xilinx ISE tool is as follows:

Speed Grade : -7

Minimum period: 4.448ns (Maximum Frequency: 224.840 MHz)

Minimum input arrival time before clock: 1.418ns

Maximum output required time after clock: 3.293ns

Clock period: 4.448ns (frequency: 224.840 MHz)

Total number of paths /destination ports: 68988 / 30564

Delay: 4.448ns (Levels of Logic = 12)

6 Conclusion

A new algorithm and architecture for Gaussian based color image enhancement system for real time applications has been presented in this paper. The Gaussian convolution used in this scheme not only smoothes the image but also removes the noise present in the image. Further, the pixels are processed in log-domain in order to compress the dynamic range. Finally, linear operation such as gain/offset correction is used in order to obtain image in display range. The Verilog design of the color image enhancement system is implemented on Xilinx Virtex-II Pro XC2VP40-7FF1148 FPGA device and is capable of processing high resolution videos up to 1600×1200 pixels at 117 frames per second. The implementation was tested with various images and found to produce high quality enhanced images. The design realized using RTL compliant Verilog fits into a single FPGA chip with a gate count utilization of about 321,804.

References

1. Anil K. Jain.:Fundamentals of Digital Image Processing, Prentice-Hall, Inc. Upper Saddle River, NJ, USA (1989).
2. An, C., and Yu, M.:Fast Color Image Enhancement based on Fuzzy Multiple-scale Retinex, In: Strategic Technology (IFOST-2011), 6th International Forum, Vol. 2, pp. 1065-1069, (2011)
3. Hines, G., Rahman, Z., Jobson, D., and Woodell, G.:Single-scale Retinex using Digital Signal Processors, In: Proceedings of Global Signal Processing Conference, Vol. 27, Citeseer, (2004)
4. Hines, G., Rahman, Z., Jobson, D., and Woodell, G.:DSP Implementation of the Retinex Image Enhancement Algorithm, In: Proceedings of SPIE Visual Information Processing XIII, Vol. 5438, (2004)
5. Zhang, M., Seow, M., and Asari, V.K.:A High Performance Architecture for Color Image Enhancement using a Machine Learning Approach, International Journal of Computational Intelligence Research-Special Issue on Advances in Neural Networks, vol. 2, no. 1, pp. 40-47, (2006)
6. Tsutsui, H., Nakamura, H., Hashimoto, R., H., Okuhata, and Onoye, T.:An FPGA Implementation of Real-time Retinex In: World Automation Congress (WAC2010) Video Image Enhancement, 2010, pp. 1-6, (2010)
7. Alsuwailem A., and Alshebeili, S.:A New Approach for Real-time Histogram Equalization using FPGA, In:IEEE Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, (ISPACS 2005), pp. 397-400, (2005)
8. Zhang, M., Seow, M., Tao, L., and Asari, V.K.:Design of an Efficient Architecture for Enhancement of Stream Video Captured in Non-uniform Lighting Conditions, In: International Symposium on Signals, Circuits and Systems, (ISSCS 2007), Vol. 2, pp. 1-4, (2007)
9. Ngo, H., Zhang, M., Tao, L., and Asari, V.K.:Design of a Digital Architecture for Real-time video, Enhancement based on Illuminance-Reflectance Model, In: 49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2006), Vol. 1, pp. 286-290, (2006)
10. Bidarte, U., Ezquerria, J., Zuloaga, A., and Martin, J.:VHDL Modeling of an Adaptive Architecture for Real-time Image Enhancement, In: Fall VIUF Workshop, pp. 94-100, (1999)
11. Tao L., and Asari, V.K.:Modified Luminance based MSR for Fast and Efficient Image Enhancement, In: 32nd Proceedings Applied Imagery Pattern Recognition Workshop, pp. 174-179, (2003)
12. Cheng, H., and Shi, X.:A Simple and Effective Histogram Equalization Approach to Image Enhancement,Digital Signal Processing, Vol. 14, No. 2, pp. 158-170, (2004)
13. Jobson, D., Rahman, Z., and Woodell, G.A:A Multiscale Retinex for Bridging the gap between Color Images and the Human Observation of Scenes, IEEE Transactions on Image Processing, Vol. 6, No. 7, pp. 965-976, (1997)
14. Shen C., and Hwang, W.:Color Image Enhancement using Retinex with Robust Envelope, In: 16th IEEE International Conference on Image Processing (ICIP-2009), pp. 3141-3144, (2009)
15. Ramachandran, S.: Digital VLSI Systems design: A Design Manual for Implementation of Projects on FPGAs and ASICs using Verilog. Springer Verlag, first ed., (2007)